

LISTING OF THE CLAIMS

This listing of claims will replace all prior versions, and listings, of claims in the application:

1. (Currently Amended) A method for a device to download a device program from a non-volatile data source of the device to a volatile memory of the device while connected to a host, comprising the following steps performed by the device:
connecting to a host; and
in response to one of a device power up and or to a device reset, as the case may be:
waiting for a first request signal from the host;
responding to the first request signal with a first negative acknowledgement (NAK);
initiating downloading the device program from the non-volatile data source of the device to the volatile memory of the device for a predetermined time period based on the request signal;
and
in response to a subsequent request signal from the host,
 - (a) if the device program is not completely downloaded, sending a subsequent NAK and continuing to download the device program from the non-volatile data source of the device to the volatile memory of the device, and
 - (b) if the device program is completely downloaded, responding to the subsequent request signal by executing the device program.
2. (Currently Amended) The method of claim 1, the non-volatile data source comprising non-volatile memory.
3. (Original) The method of claim 2, the non-volatile memory is at least one of an Electrically Erasable Programmable Read Only Memory (EEPROM) and a flash memory.

4. (Original) The method of claim 2, further comprising reading a signature from the non-volatile memory and validating the signature prior to connecting to the host.
5. (Original) The method of claim 2, further comprising reading descriptor information from the non-volatile memory prior to connecting to the host.
6. (Previously Presented) The method of claim 1, further comprising setting a pointer for tracking device program data downloaded from the data source.
7. (canceled)
8. (Previously Presented) The method of claim 1, further comprising updating a download point each time the predetermine time period is completed.
9. (Previously Presented) The method of claim 8, wherein the predetermined time period is a first time period for a data request signal, and a second time period for a status request signal.
10. (Previously Presented) The method of claim 9, the first time period is five hundred milliseconds and the second time period is fifty milliseconds.
11. (Original) The method of claim 1, the host is a USB host and the device is a USB device.
12. (Original) The method of claim 1, the predetermined time period is monitored by a timer.
13. (Previously Presented) The method of claim 1, further comprising determining and downloading a number of device program data blocks to be downloaded based on the predetermined time period.

14. (Original) The method of claim 13, the number of data blocks to be downloaded being further based on at least one of a download data rate and a block size.

15. (Original) The method of claim 13, further comprising setting a loop counter based on the number of data blocks to be downloaded.

16. (Currently Amended) A method for a Universal Serial Bus (USB) device to download firmware from a non-volatile data source of the USB device to a volatile memory of the USB device while connected to a USB host, comprising the following steps performed by the USB device:

connecting to a USB host; and

in response to one of a device power up or to and a device reset, ~~as the case may be~~:

waiting for a request signal from the USB host;

responding to a first request signal with a negative acknowledgement (NAK);

initiating downloading data blocks associated with firmware from [[a]] the non-volatile data source of the USB device to the volatile memory of the USB device based on a predetermined time period associated with the request signal type; and

in response to a subsequent request signal from the USB host,

(a) if the firmware is not completely downloaded, sending a subsequent NAK and continuing to download data blocks associated with the firmware from the non-volatile data source of the USB device to the volatile memory of the USB device, and

(b) if the firmware is completely downloaded, responding to the subsequent request signal by executing the firmware.

17. (Original) The method of claim 16, further comprising updating a download pointer that tracks the last data block downloaded.

18. (Original) The method of claim 16, further comprising sending a negative acknowledgement (NAK) to the host to intentionally postpone the transmission of the response to the request signal.
19. (Original) The method of claim 16, further comprising determining the signal request type and setting the predetermined time period accordingly.
20. (Original) The method of claim 16, further comprising determining a number of data blocks to be downloaded based on the predetermined time period.
21. (Original) The method of claim 20, the number of data blocks to be downloaded being further based on at least one of a download data rate and a block size.
22. (Currently Amended) A universal serial bus (USB) compatible device, comprising:
 - a non-volatile memory having firmware stored therein;
 - a microcontroller unit (MCU)
 - a volatile memory; and
 - an instruction memory storing instructions for execution by the MCU upon reset, the execution of the instructions controlling the device to respond with a negative acknowledgement (NAK) in response to a request signal from a host controller, to download the firmware to the volatile memory for use by the MCU for a period of time after responding with the NAK, and to continue to respond with NAKs and to download the firmware until downloading of the firmware to the MCU volatile memory has completed, and to execute the firmware in response to a request signal thereafter.
23. (Original) The device of claim 22, further comprising a memory for storing a download pointer to track the firmware download.

24. (Original) The device of claim 22, the MCU downloads data blocks associated with the firmware for a predetermined time period based on the request signal type from the host controller.
25. (Original) The device of claim 24, the MCU further determines the number of data blocks to be downloaded for the predetermined time period based on download data rate and a block size.
26. (Original) The device of claim 24, the predetermined time period is a first time period for a signal with a data stage and a second time period for a signal without a data stage.
27. (Previously Presented) The device of claim 26, the first time period is five hundred milliseconds and the second time period is fifty milliseconds.
28. (Original) The device of claim 23, further comprising a timer for monitoring the firmware download.